

A DEVELOPMENT PLATFORM FOR WIRELESS INTERNET CONNECTED ROBOTIC DEVICES*

*Rathika Rajaravivarma
Department of Computer Science
Central Connecticut State University
New Britain, CT 06050*

*Leo Cetinski
Central Connecticut State University
New Britain, CT 06050*

ABSTRACT

This paper addresses the demands of project based educational need at Central Connecticut State University (CCSU) to stay up to date with the technological developments in the ever-changing wireless communication field. The targeted audiences are from the ABET accredited Computer Science program students in their senior year of the undergraduate program and/or the students in the graduate program with concentration in networking or communication. This project provides a platform for a hands-on environment that integrates the concepts in the topic areas such as computer architecture, communication, networking, and programming. The purpose of this paper is to outline the hardware and software architectures necessary for a remote robotic car to be controlled over the Internet. It serves as a building block to describe a general remote robotic architecture that can be used to support a variety of remote robotic applications. In this paper, a project consisting of building a remote control model car that can be controlled through a wireless internet connection is illustrated. A person sitting at a host PC will be able to drive the car by clicking on controls displayed on the screen. The controls will allow the person to steer and accelerate the vehicle. The Internet connected robotic device is built using the Internet technologies and Intel's Stargate board. The architecture designed in this project can be used to support the most demanding applications requiring multimedia functionality such as live video.

* Copyright © 2005 by the Consortium for Computing Sciences in Colleges. Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CCSC copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Consortium for Computing Sciences in Colleges. To copy otherwise, or to republish, requires a fee and/or specific permission.

INTRODUCTION

The number of applications requiring Internet connected robotic devices is increasing rapidly. Industries ranging from consumer applications like home monitoring and control to commercial applications such as hazardous chemical handling require remote robotic control. In this changing world it is imperative to prepare the computer science students for the change in technology. The project discussed here reinforces the concepts in the topic areas such as computer architecture, communication, networking, and programming and provides a means to integrate them.

Implementing this concept is made possible through remote robotic applications. Advances in technology has greatly simplified and reduced the cost of developing a remote robotic device. To the exiting robotic applications adding wireless and multimedia functionality to build remote robotic devices are becoming ever more popular. A remote robotic device allows a user to control an object or sense data in a remote location over a network such as the Internet. A remote robotic device consists of a single board computer (SBC) connected to additional circuit boards, sensors, motors, and actuators.

SBC's are small computers that fit on a single circuit board and provide most or all the functions of a personal computer. They are the brains of the robot controlling the other devices and providing communication with the network. They also may perform data compression to decrease the amount of bandwidth needed across the network. SBC's contain a processor, main memory, disk memory that contains the file system, and a number of ports including serial, parallel, USB, Flash, PCMCIA, RJ45, and I2C. SBC's for remote robotic devices do not need PS2 and video ports since they do not directly interface with users. There are many manufacturers that build SBC's and are customized to support specific applications. Intel's Stargate SBC was the choice made for this project.

A robotic device also contains additional circuit boards, sensors, motors, and actuators depending upon the application at hand. Add-on circuit boards can communicate with the SBC or to each other through serial ports or through the I2C interface. Motors and actuators convert electrical power to mechanical motion and allow things to move or open and close. Sensors measure physical properties and convert these measurements to electrical signals. There are a wide array of sensors today and are used in a variety of applications. There are sensors that measure weather characteristics including temperature, barometric pressure, rain fall, and wind speed. There are ultraviolet sensors that can detect heat given off from fire. Cameras and microphones can also be considered sensors that measure light and sound. Sensors allow users to a remotely monitor a physical property at a distant location.

Remote robotic device architecture is dependent on the application at hand. Having a powerful multimedia SBC to only measure temperature over a wireless network is overkill. Instead a simple circuit board combining wireless NIC and temperature sensing capabilities is a much cheaper alternative. However, such circuit boards are rare to find and will most likely have to be custom built. A general picture of remote robotic device architecture is shown in Figure 1.

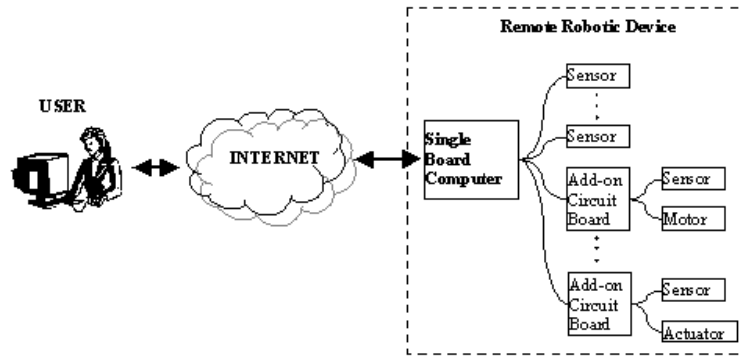


Figure 1. General Remote Robotic Device Architecture

The SBC sends sensory data to the user over the Internet. Such data can consist of many things such as the number of food items in a kitchen, security warnings, images, sound, or other sensory data. The user can send commands over the Internet to the SBC to take sensory readings, control motor speeds, or set actuator positions. Internet technologies allow any type of data to be transferred between user and remote robotic device. The increased bandwidth transmission rates permit multimedia data to be transferred at acceptable rates.

HARDWARE ARCHITECTURE

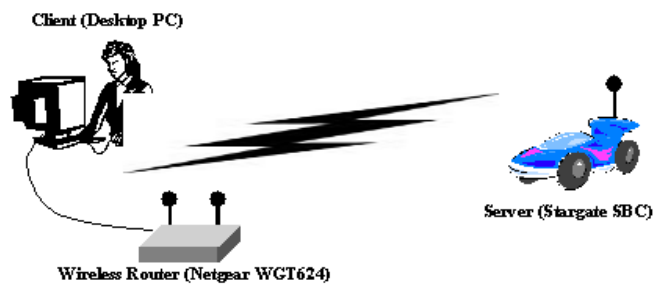


Figure 2. Wireless Network

The project hardware shown in Figure 2 consists of building a wireless computer network based on the TCP/IP model and will contain a client PC, wireless router and remote control car. The client PC is any regular desktop or laptop computer that can run a web browser. The user will control the model car through the web browser. The client PC is connected to a wireless router that will communicate to the car using 802.11b Ethernet. The wireless router is a Netgear WGT624. This wireless network allows the car to travel up to 150 feet from the router and transmit data at speeds up to 11 mbps. The model car consists of a typical R/C (remote control) car kit that can be bought at any hobby store. The kit includes the car's chassis, body, wheels, steering mechanism, gears, and electric motor. In addition to the kit, the car also contains additional items that must be purchased separately. These items include a USB camera, a SBC, a wireless flash card, an electronic speed controller, an additional circuit board, a steering servo, and a battery. A USB camera is mounted on front of the car to provide live video to the user over the Internet and is capable of providing 640x480 pixel size images at 30 fps. The SBC is mounted inside the car and is the brains of the model car. It controls the car depending on commands received from the user. The wireless flash card plugs in to the SBC and provides 802.11g wireless Ethernet connectivity. The electronic speed control provides the proper A/C current to the electric motor. The steering servo is an actuator that control's the car's steering. An additional circuit board is needed to provide the proper PWM signals to the steering servo and electronic speed controller. A 7.2-volt battery provides power to all components within the car. Figure 3 shows the overall hardware architecture of the car. More information on the items needed to build the model car is given in Table 1.

The SBC is the brain of the car. For the purpose of this project a special, single board computer called Stargate, a product of the Crossbow Company [1], was chosen. Stargate serves as Intel's newest research platform for robotics development [2]. Stargate is a popular product used for robotics development and delivers industry leading processing power with built-in multimedia capabilities for high performance and hand-held functionality. The board measure 3.5" by 2.5" yet is capable of providing rich multimedia functionality.

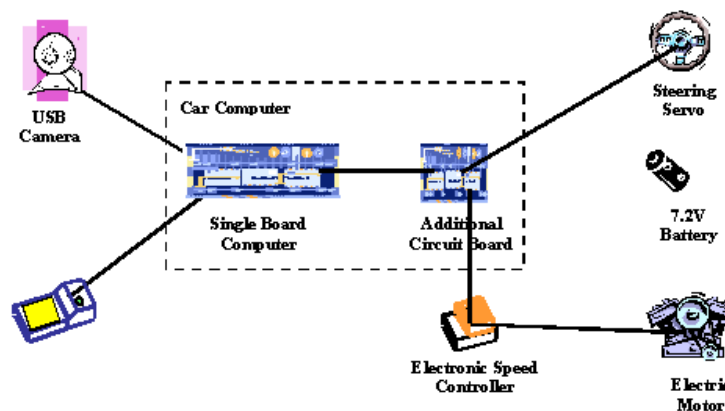


Figure 3. Hardware Modules Located Within Model

Table 1. Project Hardware Modules

Module	Company	Product	Price	More Information
Remote Control Car				
Model Car kit (includes electric motor)	Tamiya	Mad Bull	~\$92	http://www2.towerhobbies.com/cgi-bin/wti0001p?&I=LXGA39&P=0
Stargate SBC (custom made for Garcia robot)	Xbow	Stargate	\$800	http://www.acroname.com/robotics/parts/R231-SP-KIT400.html
Additional Circuit Board	Acroname	GP1.0	\$79	http://www.acroname.com/robotics/parts/S1-GP-BRD.html
Electronic Speed Controller	Duratrax	Sprint ECS	\$50	http://www2.towerhobbies.com/cgi-bin/wti0001p?&I=LXAMN7&P=0
Steering Servo	Futaba	FPS148	\$14	http://www2.towerhobbies.com/cgi-bin/wti0001p?&I=LXH318&P=0
6-Cell 1500mAh battery (7.2V)	Duratrax	Shark	2x\$11	http://www2.towerhobbies.com/cgi-bin/wti0001p?&I=LXAFD2&P=
DC Peak Charger NiCd/NiMH	Duratrax	Piranha	\$20	http://www2.towerhobbies.com/cgi-bin/wti0001p?&I=LXHB6&P=0
Wireless Flash Card	Netgear	MA701	\$56	http://www.netgear.com/products/details/MA701.php
USB Camera	Creative Labs	Webcam Pro	\$40	http://us.creative.com/products/category.asp?category=218&subcategory=219
Extra Networking Hardware				
Wireless Router	Netgear	WGT624	\$71	http://www.netgear.com/products/details/WGT624.php

As shown in Figure 4, Stargate contains a microprocessor, an I/O processor, main memory, disk memory, and multiple ports. It can provide all the functionality of a personal computer and has Linux embedded on it. However, there are no PS2 ports for a keyboard/mouse or video ports for a monitor. Communication takes place over the serial or Ethernet ports. A picture of the board is given below in Figure 4.

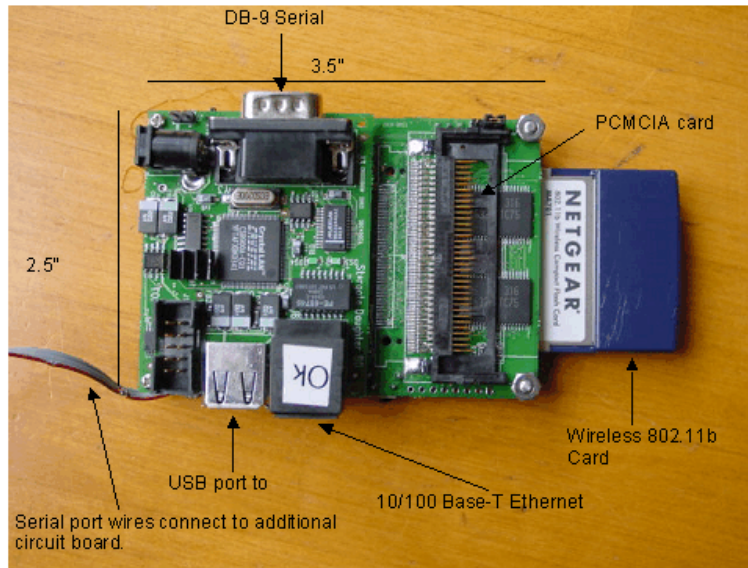


Figure 4. Stargate Board

The DB-9 Serial port can be used to connect a console to the board. Terminal emulation programs running on a host such as minicom on Linux or Hyperterminal on Windows can be used to communicate with the board. The Stargate board also contains an RJ-45 jack for 10/100 Base-T Ethernet connectivity, a PCMCIA card slot, a compact Flash card slot, a USB port, a 51-pin MICA2/GPIO, and optional I2C port. The specifications for the board are given below. It should be noted the Stargate board is made up of 2 circuit boards; a microprocessor board, and a daughter card that can be screwed together. The daughter card provides extra ports to the processor board.

STARGATE BOARD SPECIFICATIONS

STARGATE Processor Board

Intel PXA255, Xscale® 400 MHz, RISC Processor

Intel SA1111, StrongARM® Multiple I/O Companion Chip

Memory

64 MB SDRAM

32 MB FLASH Linux Software < 10 Mbytes

STARGATE Daughter Card

Communications

10 Base-T Ethernet Port RJ-45 Connector

RS-232 Serial Port DB-9 Connector

JTAG Debug Port

USB Host Port Version 1.1

General

A/C Power Adaptor 5-6 VDC, 1 Amp

Reset Button

Real-Time Clock

Communications

PCMCIA Slot Type II
 Compact Flash Slot Type II
 51-pin MICA2/GPIO UART, SSP via MICA2 Connector
 Optional I2C Port Installable Header
 Optional Serial Port (2) Installable Header

Physical

(in) 2.49 x 2.00 x 0.60
 Weight (oz) 1.42
 (g) 40.16

Environmental

0 to +70 (°C) Operating

Physical

(in) 3.50 x 2.49 x 0.73 Excluding Daughter Card
 Weight (oz) 1.68
 (g) 47.47

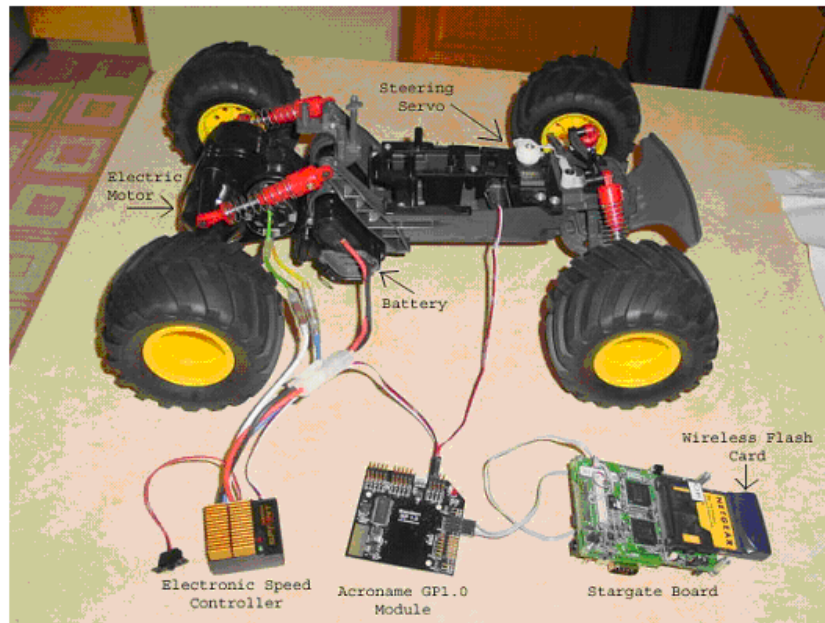


Figure 5. Remote Control Car and Components

The Stargate board makes it possible to connect a robotic device to an IP network and provide multimedia functionality. Its superior processing capabilities, reduced size, and small power consumption make it a strong contender to solve a wide array of applications. A full picture of the remote control car and components are shown in Figure 5.

SOFTWARE ARCHITECTURE

The project software consists of running a web server on the Stargate board and a web browser on the client machine. Apache is used for the web server, which runs on top

of the Linux operating system installed on Stargate. Modzilla is used for the web browser and runs on top of Linux installed on the client PC. A web page containing HTML, located on the server, is loaded in to the client's browser using the HTTP protocol. The web page displays controls for the user to control the remote car. It also displays a live video feed from the camera mounted on the car. A snap shot of the web page is shown in Figure 6.

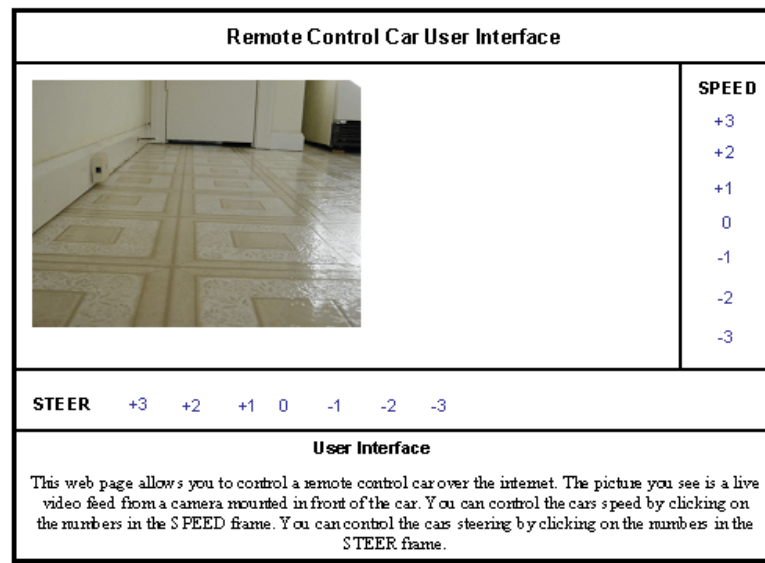


Figure 6. Car's Web Page

When the user clicks on a control setting on the web browser, it commands the web server to execute a CGI (common gateway interface) script. The CGI script sets the car's speed and steering. There are two CGI scripts located on the web server that are both written in C. One script continuously sends images from the car's camera to the client's browser. The other CGI script is used to control the car's motor and steering actuator. *The HTML for the web page and C code for the CGI scripts are the only programming that is needed for project.* The HTML is simple to code and contains simple hyperlinks that point to CGI scripts located on the server. The CGI scripts are more complicated to program. The CGI script that continuously streams images must be able to retrieve images from the web camera and perform some kind of compression for increased performance. Extra C libraries are needed to perform the image conversion and compression. Most of the code for the camera script can be obtained from the examples that come with the Stargate board. The camera script continuously runs on the web server. The CGI script that controls the electric motor and steering actuator must send the proper commands to the additional circuit board. C libraries are provided with the additional circuit board that implements these commands. Example source code is also given with the additional board to show how to run the commands. The motor/actuator script only runs when the user clicks on a control setting on the web page. The software architecture for the project is shown in Figure 7.

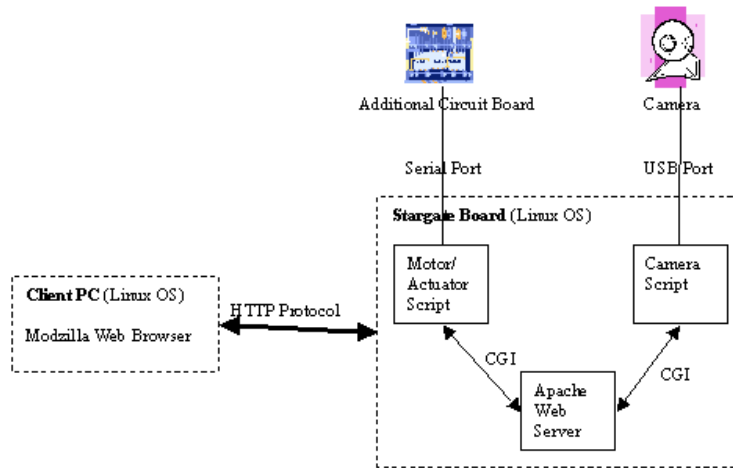


Figure 7. Project Software Architecture

STEP BY STEP INSTRUCTIONS TO BUILD PROJECT

PART I: SETUP THE STARGATE BOARD

1. **Order the items listed in Table 1.**
2. **Put together Stargate Development Kit.** Follow the instructions outlined in the manual on the CD that comes with the kit. The manual describes how to put together the kit, power it up, and communicate with it. Ensure that the LED's on the processor board light up when the board is powered on. Communication to the board can be performed through a serial cable plugged in to the board's DB-9 serial port. The manual describes how to connect to the board using either a Windows host or a Linux Host. Since a Linux development environment is needed to compile and link programs that are run on the Stargate board, Linux was chosen for this project. The Linux minicom command establishes a console connection to the Stargate board. Logging on to the Stargate board as root with a command prompt, confirms the proper functioning of the SBC. The CD that comes with the kit also contains several applications and example programs that can be run on the Stargate board. The simple "hello world" example can be compiled and linked on the host and downloaded on to the board. The program will output a message to the console when it is run on the Stargate board.
3. **Connect the Stargate board to the network using Ethernet cabling.** Power off the Stargate board and connect to the router using regular Ethernet cabling and the RJ-45 jack on the board. Power on the board and logon as root through the console. Run the ifconfig command and make sure eth0 has an IP address. Next, try to ping the board from the host computer. If successful, then the Stargate board is reachable from the network. SSH can now be used to logon to the Stargate board remotely, and SCPY can be used to copy files to and from the host.

- 4. Connect the Stargate board to the network wirelessly.** Connect a wireless NIC card to the PCMCIA or compact flash slot of the Stargate board. Boot up the board and login as root through the console port using Linux minicom. Run the command, `cardctl ident` on the Stargate board. The output of this command should show the wireless card properties. If no properties are shown, then the Stargate board is not recognizing the wireless card and the driver for the wireless card will have to be manually added. To configure the wireless card run the `iwconfig` command. This will provide the name of the interface the wireless NIC is connected to. Next run the command

```
iwconfig (interface name) essid (wireless network name).
```

This will set the correct name of the wireless network on the wireless NIC. To set the mode, run the command

```
iwconfig (interface name) mode (mode type).
```

Set mode type to “managed” if there is an access point on the network. Otherwise, set mode type to “ad-hoc”. If there is no DHCP service running on the network the command

```
ifconfig (interface name) (IP address)
```

will set the IP address of the wireless NIC. Use the `iwconfig` and `ipconfig` commands to set the parameters of the wireless network. Once the wireless NIC has been configured properly, it should be able to communicate with other hosts on the network. This can be checked with the `ping` command. Detailed instructions for connecting the Stargate board to wireless network can be found at http://www.acroname.com/intel/xscale/howto/sg_wireless.html.

- 4. Run Apache Server on the Stargate board.** The Stargate Development Kit CD contains a version of Apache that will run on the Stargate board. Follow the installation instructions to run the server on the board. Once apache is running, a sample web page from the server can be displayed in the host’s browser.

PART II: SETUP THE GP1.0 BOARD

- 1. Connect GP1.0 board to a power supply and servos.** Connect the GP1.0 board to the 7.2-volt battery. There are 2 power plugs on the board, one to power the board logic and the other to power any servo’s connected to the board. You will need to create the power jacks using the crimps kit obtained from Acroname. Create 2 power jacks and connect the board to the 7.2-volt battery. Next, connect any servos to the GP1.0 board using the servo ports on the board. There are 4 servo ports on the board.
- 2. Connect GP1.0 board to host computer with male/female DB-9 serial cable.** Use the serial interface connector and a male/female DB-9 serial cable to connect the GP1.0 board to a host computer. First connect the GP1.0 board to the host computer to make sure everything is working on the board. Then connect the GP1.0 board to the Stargate board instead of to the host PC.

3. **Download and install GP1.0 development software from Acroname.** Acroname provides example programs, documentation, and C libraries for the GP1.0 board. They can be found at - <http://www.easierrobotics.com/cgi-bin/download.cgi> [7]. Example programs are compiled and run on the host PC and show what commands to execute to properly communicate with the GP1.0 module. The Stub example shows the commands needed to control the servo's connected to the GP1.0 board. (The example needs to be edited slightly in order for it to work properly). Once the Stub example functions properly, and can control the servo positions, we know the GP1.0 board is functioning properly.
4. **Disconnect the GP1.0 board from the host and connect to the Stargate Board.** Once we know that both boards are functioning properly, we can connect them together. Disconnect the GP1.0 board from the host PC and disconnect the serial interface connector. Plug in the serial cable extending from the Stargate board into the GP1.0 board in place of the serial interface connector. Connect the Stargate board to the host PC (using either ssh or minicom). Power on both boards. Logon to the Stargate board and download the Stub example on to it. Run the Stub example on the Stargate board, to make the servos connected to the GP1.0 board move.

PART III: BUILD CAR WEB PAGE AND CGI SCRIPTS

1. **Create the motor/actuator CGI script.** The next step is to convert the Stub example in to a CGI script to control the motor and steering servo. A CGI script needs to be able to accept parameters from the web server and tell the server what to return back to the client browser. It receives parameters from the server by accessing the global variable, QUERY_STRING. This variable contains the values for setting the servos to the proper positions. The script will then execute commands as given in the Stub example to communicate with the GP1.0 board. Lastly, the script will output the text, "Location: (previous web page url)", to standard output. This will command the server to direct the client browser back to the web page from where the hypertext was clicked. The CGI script is the Stub example which has been updated to communicate with the apache web server.
2. **Create the camera CGI script.** The Stargate Development Kit CD contains example code for grabbing images from a web camera. Simply compile, link, and install this software on the Stargate board. Use PPM images so that additional libraries for image conversion are not necessary. Connect a USB camera to the Stargate board. Once the code is working and can successfully grab an image, convert it to a CGI script as done in step 1. This script will continuously run in a loop, grabbing images and outputting them to standard output.
3. **Create the car control web page.** The main car web page contains 4 frames. Each frame points to a different web page. The picture web page contains an image element that links to the camera CGI script. The speed control and steering control web pages contain hyperlinks that each link to the motor/actuator CGI script. Each hyperlink passes a different set of parameters to the motor/actuator CGI script. The different values determine the speed and direction of the car. The instructional web page contains text that describes how to control the car. Once the CGI scripts and

web pages are created, a user can control the car through the web browser on a host computer.

CONCLUSION AND FUTURE LAB PROJECTS

Intel's Stargate robotics development platform is an excellent platform for learning about IP connected robotic devices. Its small form factor, minimal power consumption, multimedia capabilities, and port connectivity is capable of supporting the most complex applications. The increased transmission rates of Ethernet technologies and efficient compression hardware from Intel make rich multimedia capability possible. Stargate platform employs the application of knowledge in hardware and software and can be used to learn about many types of "higher level" technologies. For example, it can be used in a course on network programming to program IP sockets to communicate directly to the client over the network instead of using a web server. In these instances the camera CGI script must be updated to output the image data to the IP socket instead of the standard output and the client program at the browser must be modified to receive the image data from the server through an IP socket. This would improve the speed and the response time for real-time communication. The remote sensing capability offers interactive solutions for ubiquitous computing. This project can also be extended for the wireless sensor network applications and in a number of location-based applications.

REFERENCES

1. <http://www.xbow.com/Products/productsdetails.aspx?sid=85> - The Crossbow website for the Stargate board. Contains specifications and data sheets.
2. <http://www.intel.com/design/intelxscale/> - This Intel web page describes the Xscale technology used on the Stargate board.
3. <http://platformx.sourceforge.net/> - This is a great web site for learning about all resources related to the Stargate board.
4. <http://stargate.sourceforge.net/> - This is a future web site that will hold information on the Stargate board.
5. <http://www.acroname.com/robotics/parts/R231-SP-KIT400.html> - The Acroname web site for the Stargate board. It has links to some great articles and other resources.
6. <http://www.cens.ucla.edu/pipermail/stargate-users/> - Stargate mailing list to post and answer problems.
7. <http://www.easierrobotics.com/cgi-bin/download.cgi> - The Acroname download site containing software and documentation for the GP1.0 board.